E-Arogya - Bug #67

Feature # 65 (New): Security Audit

[Security Audit] 2-Insecure Data Storage

17/04/2024 03:29 PM - Kalyan Battula

Status:	In Progress	**Start date:**	17/04/2024
Priority:	High	**Due date:**	
Assignee:	Srinivas Kanukolanu	**% Done:**	0%
Category:		**Estimated time:**	0:00 hour
Target version:	Security Audit	**Spent time:**	0:00 hour
Deployed In:		**Category:**	

Description

Insecure Data Storage
CWE : CWE-312
Description :
Insecure data storage vulnerabilities occur when development teams assume that users or malware will not have access to a mobile device's file system and subsequent sensitive information in data-stores on the device. File systems are easily accessible.
Organizations should expect a malicious user or malware to inspect sensitive data stores. Usage of poor encryption libraries is to be avoided. Rooting or jailbreaking a mobile device circumvents any encryption protections. When data is not protected properly, specialized tools are all that is needed to view application data.
Affected Path(s) :
https://earogya.satragroup.in/frontdesk/dashboard *-Applicable to entire application
Impact :
Insecure data may result in the following business impacts: • Identity theft; • Privacy violation; • Fraud; • Reputation damage; • External policy violation (PCI); or • Material loss.
Recommendation :
1. It is important to threat model your mobile app, OS, platforms and frameworks to understand the information assets the app processes and how the APIs handle those assets. It is crucial to see how they handle the following types of features :
1. URL caching (both request and response);
2. Keyboard press caching;
3. Copy/Paste buffer caching;
4. Application backgrounding;
5. Intermediate data
6. Logging;
7. HTML5 data storage;
8. Browser cookie objects;
9. Analytics data sent to 3rd parties.
2. Also, it is recommended to encrypt the user data in device internal storage.

Evidence/Proof Of Concept :
Step 1: Login to the application, access the internal pages and click on logout. It is observed that even after the user get logged out, the JWT token was still saved in the cookie storage as shown below.

clipboard-202404171528-m9ey7.png
Step 2: The JWT token reveals the sensitive information like username etc., as shown below.

clipboard-202404171526-tp2zu.png

Step 3: Access the above URL, here Insecure data storage as shown that.

clipboard-202404171527-ghf2c.png

History

#1 - 17/04/2024 03:54 AM - Kalyan Battula

- Subject changed from [Security Audit] Insecure Data Storage to [Security Audit] 2-Insecure Data Storage

- Priority changed from Normal to High

#2 - 23/04/2024 03:04 AM - Uma Maheswarachari Melpati

- Assignee set to Uma Maheswarachari Melpati

#3 - 23/04/2024 11:18 PM - Uma Maheswarachari Melpati

- Assignee deleted (Uma Maheswarachari Melpati)

#4 - 24/04/2024 12:41 AM - Vasudev Mamidi

- Assignee set to Srinivas Kanukolanu

#5 - 24/04/2024 12:48 AM - Vasudev Mamidi

- Status changed from New to In Progress

#6 - 03/05/2024 05:28 AM - Sivakanth Kesiraju

- Target version set to Security Audit

Files

clipboard-202404171526-tp2zu.png	94.9 KB	17/04/2024	Kalyan Battula
clipboard-202404171527-ghf2c.png	68.7 KB	17/04/2024	Kalyan Battula
clipboard-202404171528-m9ey7.png	86.3 KB	17/04/2024	Kalyan Battula